

Controlling Candidate-Sequential Elections*

Edith Hemaspaandra[†]
 Department of Computer Science
 Rochester Institute of Technology
 Rochester, NY 14623, USA

Lane A. Hemaspaandra[‡]
 Department of Computer Science
 University of Rochester
 Rochester, NY 14627, USA

Jörg Rothe[§]
 Institut für Informatik
 Heinrich-Heine-Universität Düsseldorf
 40225 Düsseldorf, Germany

February 29, 2012

Abstract

All previous work on “candidate-control” manipulation of elections has been in the model of full-information, simultaneous voting. This is a problem, since in quite a few real-world settings—from TV singing/dancing talent shows to university faculty-hiring processes—candidates are introduced, and appraised by the voters, in sequence. We provide a natural model for sequential candidate evaluation, a framework for evaluating the computational complexity of controlling the outcome within that framework, and some initial results on the range such complexity can take on. We hope our work will lead to further examination of temporally involved candidate control.

1 Motivating Example

In an author’s school, faculty hiring happens basically as follows. On some Mondays, a candidate visits, gives a talk, and meets with faculty members. Then each of the department’s rank-and-file faculty members mails to a staff member his or her ranking of all the candidates so far, namely, by inserting the new candidate into the preference order he or

*Also appears as URCS-TR-2012-975.

[†]Supported in part by grant NSF-CCF-1101452 and a Friedrich Wilhelm Bessel Research Award. Work done in part while visiting Heinrich-Heine-Universität Düsseldorf.

[‡]Supported in part by grants NSF-CCF-{0426761,0915792,1101479} and arc-dp110101792, and a Friedrich Wilhelm Bessel Research Award. Work done in part while visiting Heinrich-Heine-Universität Düsseldorf.

[§]Supported in part by DFG grant RO-1202/15-1, SFF grant “Cooperative Normsetting” of HHU Düsseldorf, and a DAAD grant for a PPP project in the PROCOPE program.

she sent after the previous candidate. The department Chair typically follows up by emailing/phoning the candidate a day or two after the visit, so that occurs after the Chair has seen the faculty rankings generated by the candidate’s visit. Moving now from reality to (slight?) fiction, let us imagine that the Chair in that followup can easily choose to scare away a candidate (“Oh, did I remember to mention that if you come your office will be a shared closet in our lovely basement, I’ll help you broaden yourself by teaching a wide range of introductory courses, and I see in you a real talent for extensive committee work which I’ll put to good use?”). But let us further assume that the Chair cannot do this more often than a certain threshold, as otherwise the rank-and-file faculty will realize the Chair is manipulating the process and will revolt. So, how should the Chair use this power of candidate suppression to most effectively ensure that one of the candidates the Chair likes will at the end of the process win the election (under the faculty preferences, among the candidates not scared away)?

This example nearly perfectly captures the topic and model of this paper. We are moving what in the literature is called “candidate control” [BTT92] (in the example, of the sort known as “constructive control by deletion of candidates”) from its existing setting of simultaneous elections into a setting where preferences are set/revealed sequentially, and the Chair right after the preferences related to an introduced candidate are revealed must use-or-forever-lose the ability to suppress that candidate.

We also are interested—again moved to a sequential setting—in constructive control by adding candidates, a natural analogue of the above, and in destructive versions of both adding/deleting candidates, which are the same issues except the Chair’s goal is to ensure that none of a certain set of hated candidates is hired.

2 Formalizing the Problem

Let us discuss how to formalize this into a decision problem whose complexity can be studied. Due to space, we’ll do so here in detail just for constructive control by deleting candidates. Let \mathcal{E} denote the underlying election system: a mapping from candidates and votes over the candidates (with preferences typically as strict, linear orderings) to a set of winners. The candidates left standing at the end will be fed into this election system along with the votes (with each vote’s preference order masked down to that set of still-standing candidates).

The input will capture a “moment of decision” for the Chair. That is, the input will give the history of the process up to the given point, and then will ask whether there is some action of the Chair that can ensure she will get a happy outcome. We must make it clear what we mean by this. We will be inspired by the recently introduced sequential approach to manipulation of [HHRa], which also centers on a “moment of decision” and that takes the same “can we do at least this well even if fate conspires against us” approach adopted below. However, that paper is completely focused on *voters* appearing sequentially; the model of *candidates* appearing sequentially with preferences set/revealed as they appear is foreign to that earlier work.

The input will be the set of candidates, the set of voters, the order in which the candidates will be presented, a flag denoting which the current candidate is, a bound k on the maximum number of candidates the Chair can suppress, an ordering σ of how the Chair views all candidates (the Chair had the c.v.'s ahead of time and has evaluated them already), a specific candidate d such that the Chair's goal is to ensure that there is an election winner from the set $\{c \mid c \geq_\sigma d\}$ (i.e., d or some candidate the Chair likes better than d is a winner), and the history up to the current moment in time (which means for each candidate before the current one a bit saying whether the Chair deleted that candidate, and a preference order for each voter over all the candidates up to and including the current one—we could also make this just over all as-yet nondeleted candidates, but let us make it over all candidates so far, though it doesn't affect the eventual results). And the question being asked in this decision problem is whether there is some decision the Chair can make about the current candidate (to delete, or not to delete) such that, assuming that the Chair at each future decision is free to act in light of the information revealed up to that point, the Chair can ensure that the winner set will have nonempty intersection with the candidates she likes, $\{c \mid c \geq_\sigma d\}$, regardless of what else happens in the election (i.e., even if the revealed preferences are highly unfavorable to the Chair's wishes).

The decision problem (i.e., language) here is simply the set of all inputs where the answer to that question is Yes. Let us call this problem *online \mathcal{E} -constructive-control-by-deleting-candidates*.

Briefly, the “adding” candidates analogue is almost the same—except the input contains a “certainly in the election” set of candidates, and a (disjoint) set of “potential additional” candidates, and a presentation ordering over the union of those two sets, and the rest is analogous (so for potential-addition candidates before the current one the input tells whether the Chair added them, etc.).

And these constructive-control adding and deleting cases each have a “destructive control” sibling, where the question is whether the Chair can ensure that no one “ d or worse” is a winner. (For destructive control by deleting candidates, there is a special issue as to whether the Chair can simply start deleting some or all candidates who are “ d or worse,” thus perhaps ruthlessly obtaining her goal. Our default model—call it the “gonzo” model—is that the Chair may delete some, but never all, of the candidates who are “ d or worse.” An alternate model—call it the “hand-tied” model—is that the Chair may never delete anyone who is “ d or worse.” The results we mention in this paper for destructive control by deleting candidates hold equally well for both those models.)

In the language of multiagent systems, candidates are alternatives and voters are agents. So though about “elections,” this model is equally well about preference aggregation in multiagent systems in which the alternatives are sequentially revealed and evaluated by the agents, and another party is trying to control the outcome.

3 Complexity Results

Let us assume that our election system's (\mathcal{E} 's) winner-determination problem (i.e., "Is candidate c a winner under this election system, if the candidates and votes are C and V ?") is in polynomial time. Then it is easy to see that all of our above online candidate control problems can be solved within the complexity class PSPACE, the well-known class of problems solvable in polynomial-space (note: $\text{NP} \subseteq \text{PSPACE}$). That is because the process is essentially about alternating quantifiers: Is there some legal move by the Chair about the current candidate, such that for all possible settings of the information revealed after this up to the Chair's next decision, there is a legal next decision by the Chair, such that... .. such that the winner set contains either d or some candidate the Chair likes more than d . The PSPACE upper bound remains valid even if we restrict \mathcal{E} 's winner problem not to P but rather to PSPACE.

Clearly, not all election systems will require the full power of PSPACE for mounting control attacks. It is easy to construct artificial systems where all these control attacks have polynomial-time control complexity. But a more important question is whether the PSPACE upper bound is itself too enormous. Can such tremendous control complexity be realized, even for election systems whose winner problems must be in polynomial time?

The answer is yes. Although the construction is not simple, we have by setting up appropriate election systems and reductions from intractable problems, shown that for each of the four problems defined above, there is an election system with a polynomial-time winner problem for which the online control problem of the given type is PSPACE-complete.

Briefly put, the construction enmeshes issues of formulas into election systems in a way that so tightly incorporates and interprets formulas, variables, and assignments, that one can by using a careful reduction and some legal preprocessing transformations ensure that the process of the online control attempt can succeed exactly if the input to a PSPACE-complete formula-problem that transformed into that problem is a positive instance.

4 Related Work

Control was introduced in the seminal paper of Bartholdi, Tovey, and Trick [BTT92], which in the bounded-rationality spirit of Simon [Sim69] made the point that computational complexity is important in decision-making. [BTT92] defined non-online versions of the constructive-deletion notion used above and a precursor of the constructive-addition notion used above. The non-online versions of the construction-addition notion used above and both destructive notions used above is from [HHR07], although destructivity had been introduced even earlier by Conitzer, Sandholm, and Lang [CSL07] for a different type of attack known as manipulation. There have been many papers analyzing the (non-online) control complexity of many election systems, and seeking to find natural systems that make many types of control attack difficult (see the survey [FHH10] and the references therein).

Our model of the process's goal, having the Chair try to guarantee a goal under the most hostile of responses, is inspired by the area of online algorithms [BE98], and was used for

online manipulation in [HHRa]. That paper adopted a point-in-time view of voter-sequential elections, as does the current paper for candidate-sequential elections. In contrast, a full-information, game-theoretic approach to voter-sequential/roll-call elections can be found in the very interesting, earlier work of Xia and Conitzer [XC10] (see also [DP01, Slo93] and the references therein), which in part inspired this work.

This paper studies online candidate control; a sister paper studies online voter control [HHRb].

5 Open Directions

Our contribution is initial results for a research direction, candidate-sequential elections, that we suggest is of interest, not as a replacement for the study of voter-sequential elections, but as a notion that captures different but also important settings.

It will be important to seek results for the complexity, in this model, of natural systems—ideally both in the worst-case and in typical-case models. Another interesting direction will be to also give the Chair limited or total control over the candidate presentation order; in political science, for example, in many settings control of agenda-order can be powerful.

References

- [BE98] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.
- [BTT92] J. Bartholdi, III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modeling*, 16(8/9):27–40, 1992.
- [CSL07] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 2007.
- [DP01] E. Dekel and M. Piccione. Sequential voting procedures in symmetric binary elections. *Journal of Political Economy*, 108(1):34–55, 2001.
- [FHH10] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Using complexity to protect elections. *Communications of the ACM*, 53(11):74–82, 2010.
- [HHRa] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The complexity of online manipulation of sequential elections. In preparation.
- [HHRb] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Online voter control in sequential elections. In preparation.
- [HHR07] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.

- [Sim69] H. Simon. *The Sciences of the Artificial*. MIT Press, 1969. Third edition, 1996.
- [Slo93] B. Sloth. The theory of voting and equilibria in noncooperative games. *Games and Economic Behavior*, 5(1):152–169, 1993.
- [XC10] L. Xia and V. Conitzer. Stackelberg voting games: Computational aspects and paradoxes. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 697–702. AAAI Press, July 2010.